Authenticated Group Diffie-Hellman Key Exchange :

Theory and Practice

Olivier Chevassut (UCL - LBNL)

J.- J. Quisquater (UCL - Promotor)

D. Agarwal (LBNL - co-advisor, U.S.A)

D. Pointcheval (ENS - co-advisor, France)

Outline

- Introduction
  - — motivation
  - — research objectives

- Background
- Contributions
- Group Diffie-Hellman key exchange
- Dynamic Group Diffie-Hellman key exchange
- Refinements
- Practical aspects
- Conclusion and further work

Motivation

- An increasing number of distributed applications need to communicate within groups, e.g.
  - collaboration and videoconferencing tools
  - replicated servers
  - stock market and air traffic control
  - distributed computations (Grids)
- An increasing number of applications have security requirements
  - privacy of data
  - protection from hackers (public network)
  - protection from viruses and trojan horses
- Group communication must address security needs

Research Objectives

- Provide reliable communication for collaborating groups spread across the Internet
  - simplify distributed application development
  - simplify communication between components in distributed applications
  - support flexible delivery capabilities to support a broad range of application needs (e.g., ordering)
- Provide a secure channel among the group members with security services (similar to SSL)
  - support confidentiality, authenticity, and integrity
  - support access control based on certificates
  - security services optional

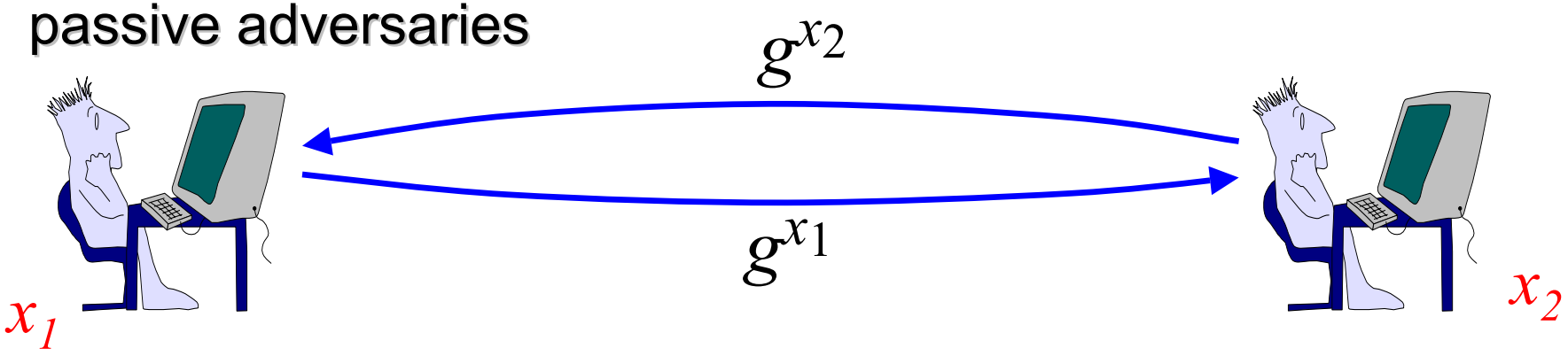
Outline

- Introduction
- > Background
  - — two-party Diffie-Hellman key exchange
  - — design methodologies
  - — how provable security works

- Contributions
- Group DH key exchange
- Dynamic Group DH key exchange
- Refinements
- Practical aspects
- Conclusion and further work

The Two-Party Diffie-Hellman Algorithm

- Establishing a secure channel between two principals is reduced to the problem of generating a session key sk
- The session key is used to achieve data secrecy and integrity
- The original DH algorithm from 1976 was only secure against passive adversaries



$$sk = g^{x_1 x_2}$$

Design Methodologies

- Ad hoc or heuristic security
  - — attack-response design not successful
  - — helps avoid known attacks
- Formal Methods [BAN90]
  - — formal specification tools
  - — successful at finding flaws and redundancy
  - — assurance limited to formal system
- Provable Security [GM85]
  - — based on complexity theory
  - — successful at avoiding flaws
  - — useful to validate cryptographic algorithms

How Provable Security works

1. Specification of a model of computation
   - instances of players are modeled via oracles
   - adversary controls all interactions among the oracles
   - adversary's capabilities are modeled by queries to the oracles
   - adversary plays a game against the oracles
2. Definition of the security goals
   - authentication, freshness and secrecy of session keys, forward-secrecy
3. Statement of the intractability assumptions
   - computational/decisional Diffie-Hellman (CDH and DDH)
4. Description of the algorithm and its proof of security
   - proof shows by contradiction that the algorithm achieves the security goals under the intractability assumptions

Contributions

1. [BCPQ01a] Authenticated GDH key exchange, ACM Computer and Communications Security, 2001

2. [BCP01b] Authenticated dynamic GDH key exchange, Asiacrypt, 2001

3. [BCP02] Refinements - forward-secrecy, Eurocrypt, 2002

4. [ACTT01] Practical aspects, IEEE Symposium on Computer and Communications, 2001

Outline

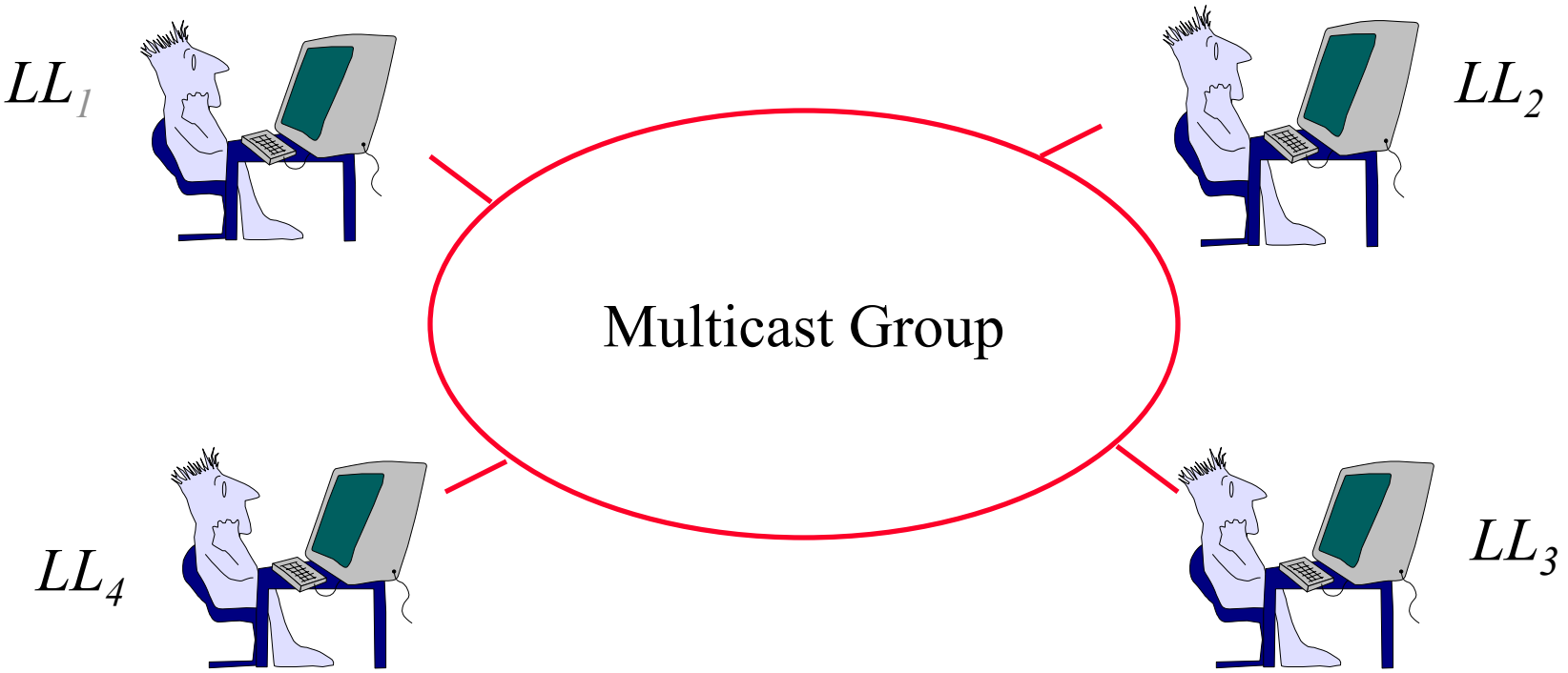
- Introduction
- Background
- Contributions
- Group Diffie-Hellman key exchange
  - model of computation
  - security goal of authenticated key exchange
  - description of an algorithm and its proof of security
- Dynamic Group DH key exchange
- Refinements
- Practical aspects
- Conclusion and further work

[BCPQ01a] Group Diffie-Hellman Key Exchange: The Setting

- Member characteristics
  - — small number of users (up to 100 members)
  - — members have similar computing power
  - — no hierarchy among members (no client/server)
  - — many-to-many communication
- Membership characteristics
  - — all members join the group at once
  - — membership participants are known in advance

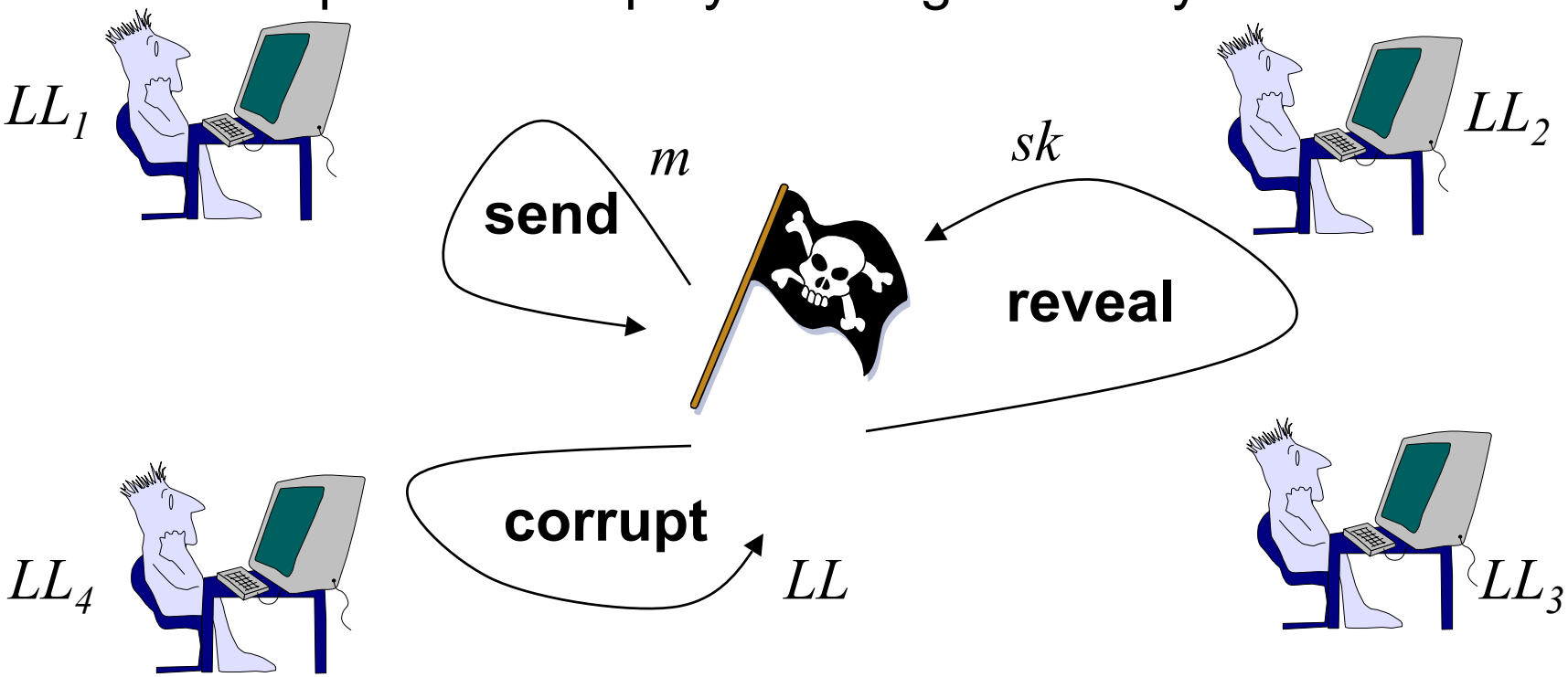
Model of Communication

- A multicast group consisting of a set of n players
  - — each player is represented by many instances/oracles
  - — each player holds a long-lived key (LL)

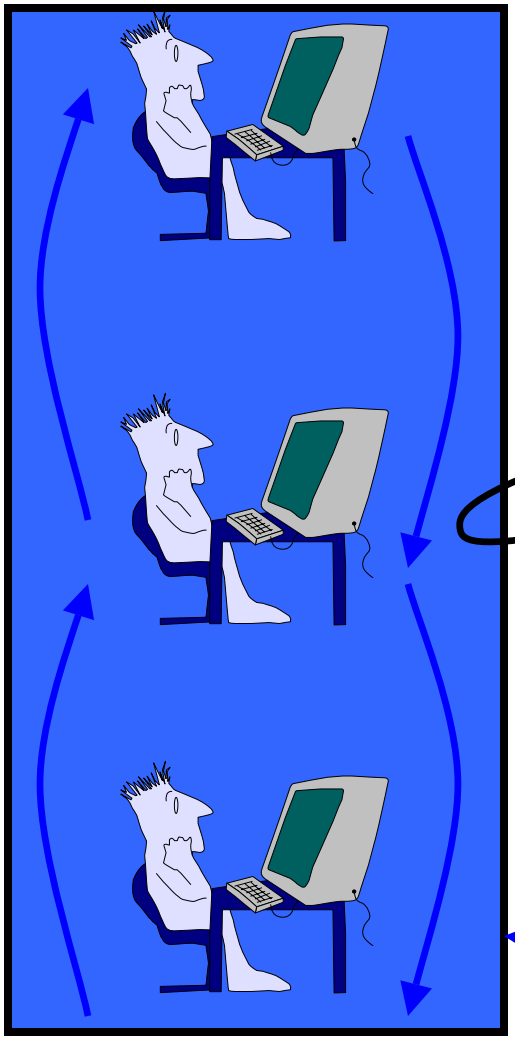


Modeling the Adversary

- Adversary's capabilities modeled through queries
  - send: send messages to instances
  - reveal: obtain an instance's session key
  - corrupt: obtain a player's long-lived key



Freshness Related Queries



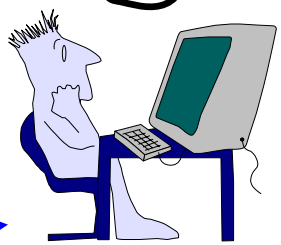
***sk* is Fresh if it is known by the players but not the adversary**

reveal

(sk)

(LL)

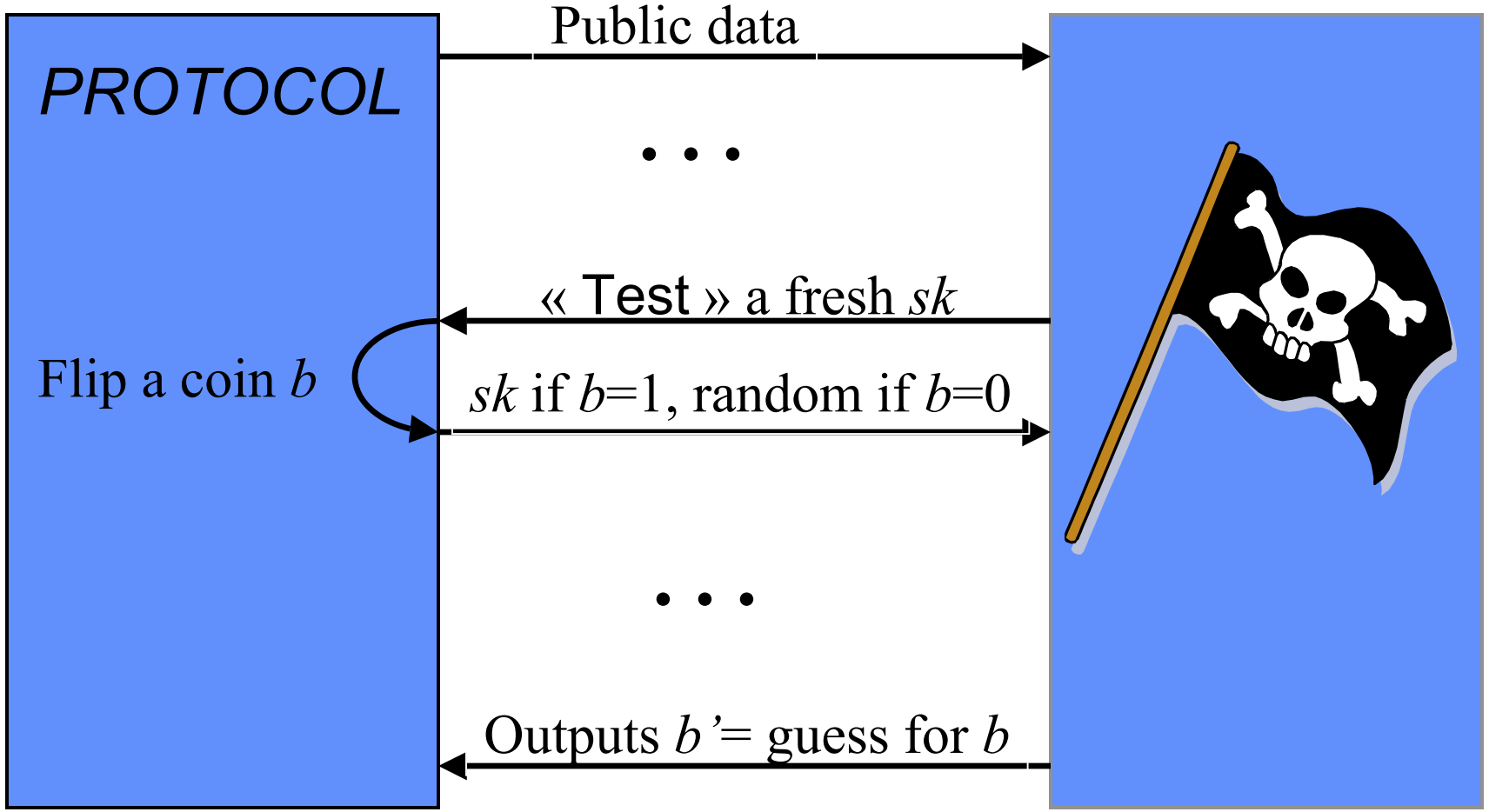
corrupt



Security Goal : AKE
Authenticated Key Exchange

- Implicit authentication
  - — Only the intended partners can compute the session key
- Semantic security
  - — the session key is indistinguishable from a random string
  - — modeled via a Test-query

Security Goal: The Game

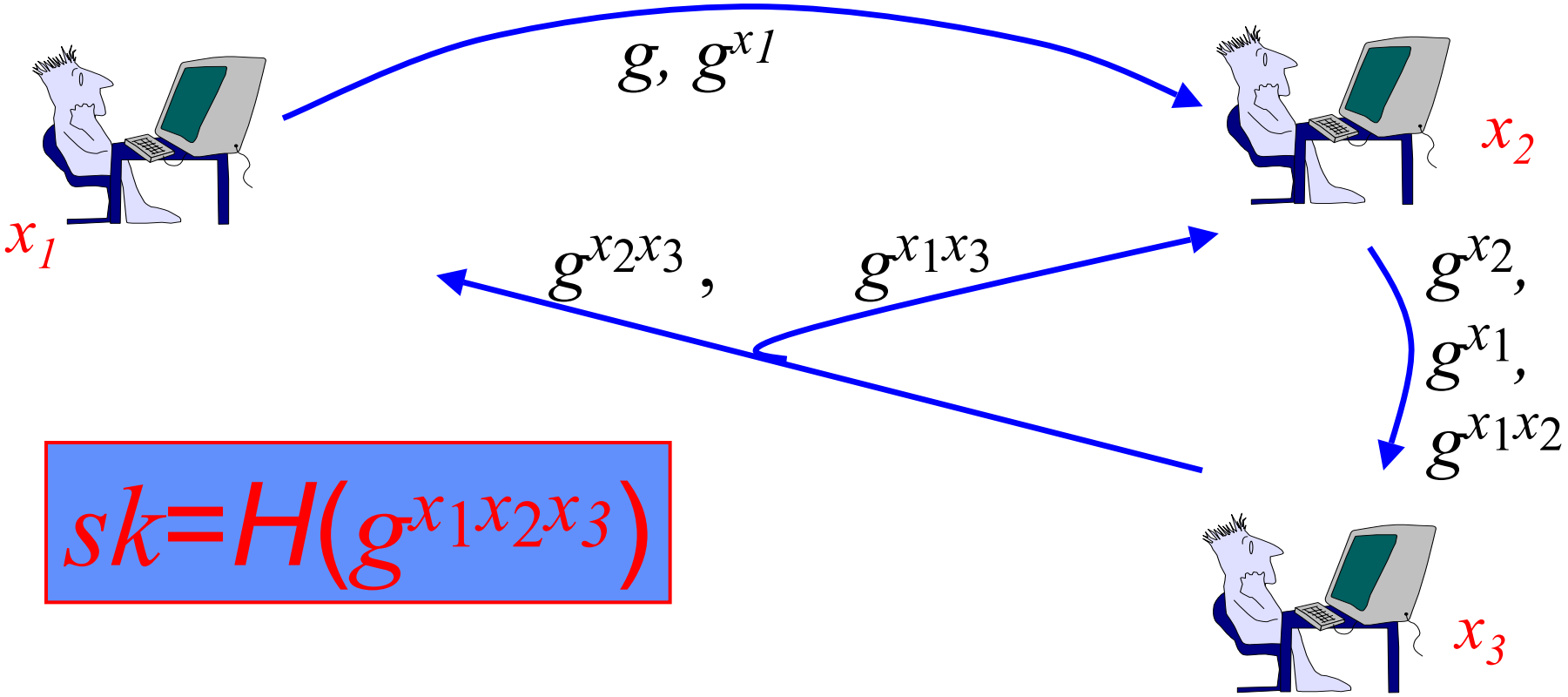


An Algorithm for Authenticated Group DH Key Exchange

- The session key is
  - $sk=H(g^{x_1 x_2 \ldots x_n})$
- Ring-based algorithm with signed flows :
  - up-flow: the contributions of each instance are gathered
  - down-flow: the last instances broadcasts the result
  - instances compute the session key from the broadcast
- Many details abstracted out

The Algorithm

- Up-flow: U_i raises received values to the power of x_i and forwards to U_{i+1}
- Down-flow: U_n processes the last up-flow and broadcasts



Security Measurement

- Using ideal-hash assumption
- Theorem

$$\mathrm{Adv}^{ake}(t,q_s,q_h) \leq n \cdot \mathrm{Succ}^{cma}(t') + 2 \cdot q_s^{n} \cdot q_h \cdot \mathrm{Succ}^{gcdh}(t'')$$

$$t',t'' \leq t + q_s \cdot n \cdot T_{exp}(k)$$

- The adversary can break the algorithm in two ways

  (1) the adversary forges a signature w.r.t some player's LL-key => it is possible to build a forger (CMA)

  (2) the adversary is able to guess the bit b involved in the Test-query => it is possible to solve an instance of the GCDH problem

Outline

- Introduction
- Background
- Contributions
- Group DH key exchange

> Dynamic Group Diffie-Hellman key exchange
  — model of computation
  — description of an algorithm and its proof of security

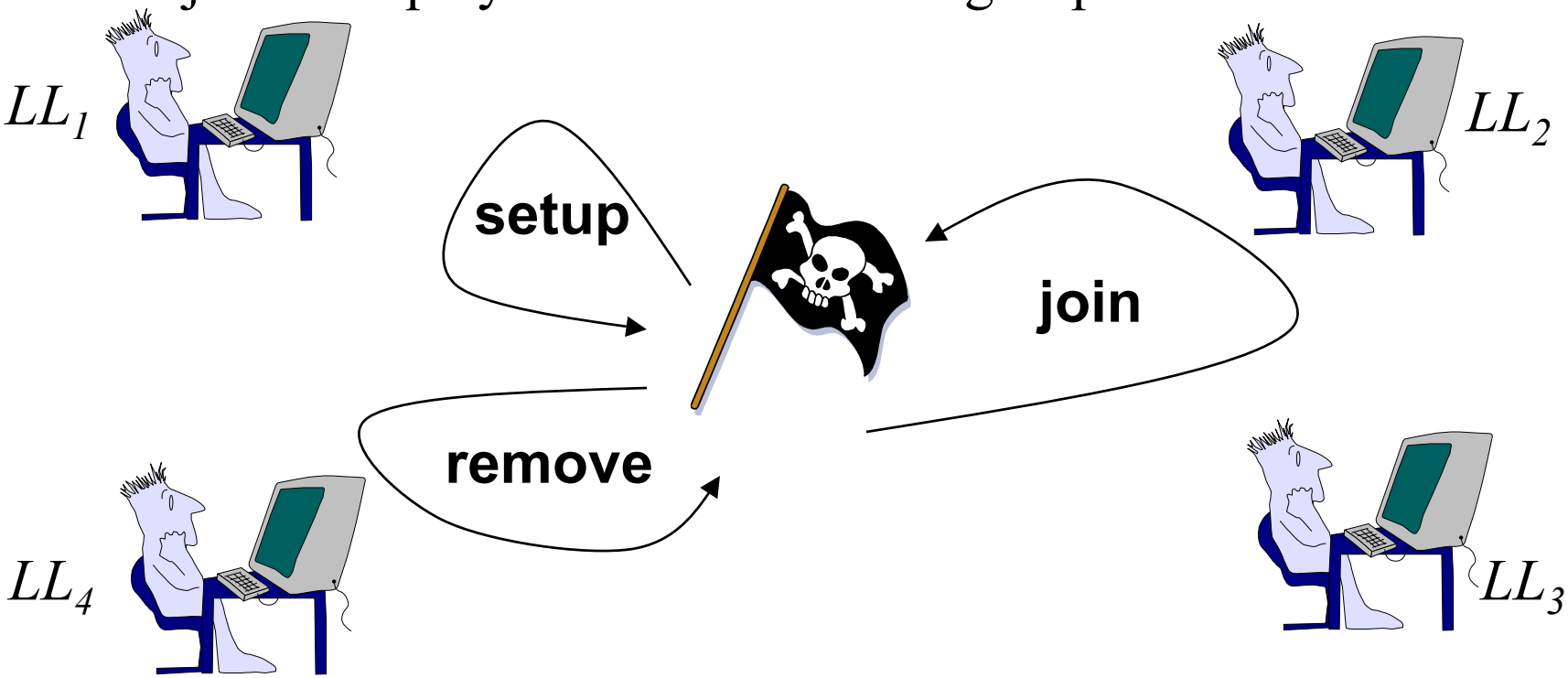
- Refinements
- Practical aspects
- Conclusion and further work

[BCP01b] Dynamic Group DH key Exchange: The Setting

- Additional membership characteristics
  - members join and leave the group at any time
  - network partitions and merges (i.e asynchronous network with failures)
  - membership is incrementally defined

Modeling the Adversary

- Adversary's additional queries
  - — setup: initialize the multicast group
  - — remove: remove players from multicast group
  - — join: add players to the multicast group

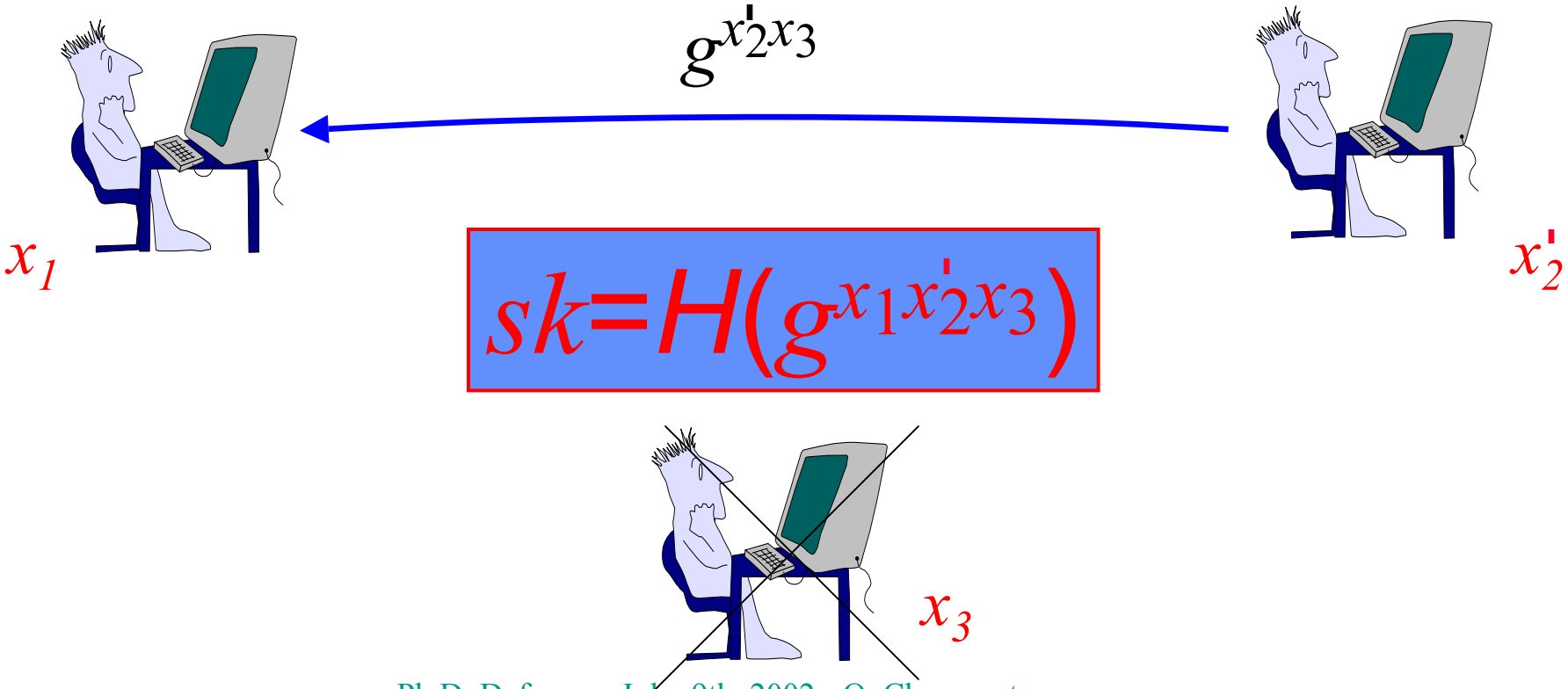


An Algorithm for Authenticated Dynamic Group DH Key Exchange

- The session key is
  - — $sk=H(g^{x1x2\ldots xn})$
- Ring-based with signed flows
- Defined by two additional algorithms
  - — JOIN
  - — REMOVE
- Many details abstracted out

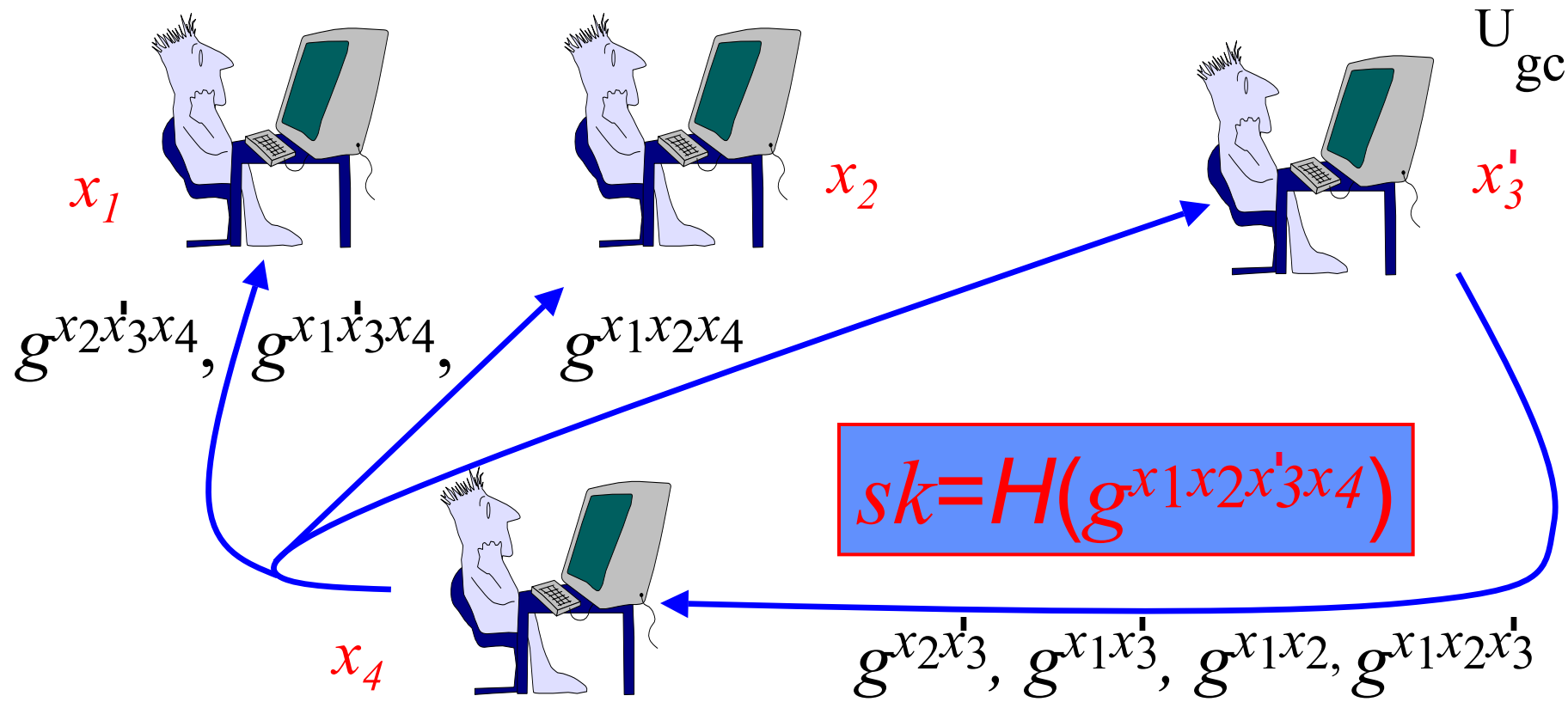
The REMOVE Algorithm

- Down-flow: player with highest index (U_{gc}) raises the previous saved broadcast to the power of its new private exponent and broadcast the result



The JOIN Algorithm

- Up-flow : U_{gc} raises the previous saved broadcast to the power of its new private exponent and forwards to U_{i+1}
- Down-flow: U_n processes the last up-flow and broadcasts



Security Measurement: Authenticated Key Exchange (AKE)

- Ideal-hash assumption
- Theorem

$$\mathrm{Adv}^{ake}(t,Q,q_s,q_h) \leq 2 \cdot n \cdot \mathrm{Succ}^{cma}(t') + 2 \cdot Q \cdot \binom{n}{s} \cdot s \cdot q_h \cdot \mathrm{Succ}^{gcdh}(t'')$$

$$t',t'' \leq t + (Q+q_s) \cdot n \cdot T_{exp}(k)$$

- The adversary can break the protocol in two ways

  (1) the adversary forges a signature w.r.t some player 's LL-key => it is possible to build a forger (CMA)

  (2) the adversary is able to guess the bit b involved in the Test-query

  => it is possible to come up with an algo that solves an instance of the GCDH problem

Outline

- Introduction
- Background
- Contributions
- Group DH key exchange
- Dynamic Group DH key exchange

> Refinements

— security goal of strong forward-secrecy

- Practical aspects
- Conclusion and further work

[BCP02] Security Goal : Strong Forward-Secrecy

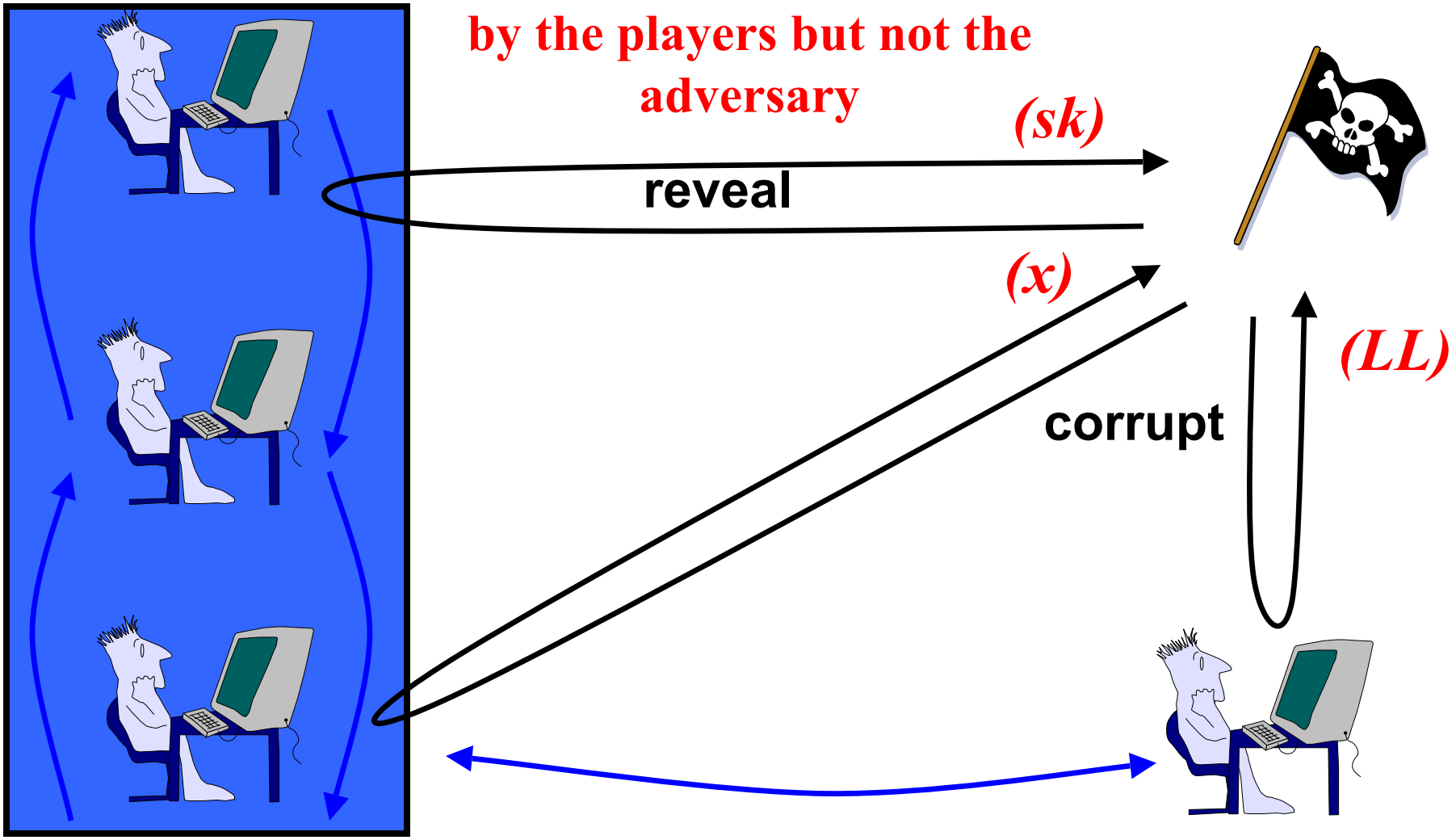
- Weak forward-secrecy :

  The corruption of a player 's long-lived key does not compromise the security of previously established session keys

- Strong forward-secrecy :

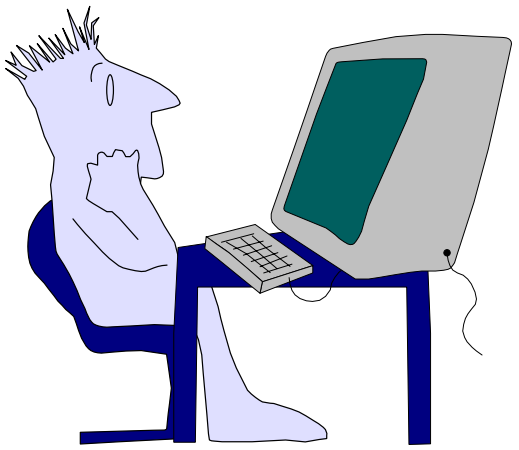
  The corruption of a player 's internal state and long-lived key does not compromise the security of previously established session keys

Freshness Related Queries

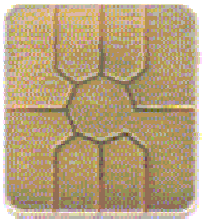


A Dynamic Group DH Key Exchange Protocol using Crypto-Devices

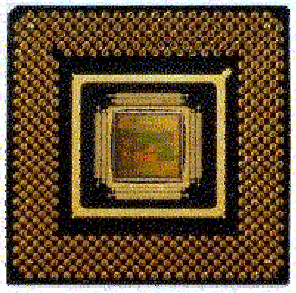
- Modifications to the algorithms to achieve strong FS
- Smart-card performs the authentication functions
- Crypto-processor performs the key exchange functions



U_i



$(s_i, g^{s_i}) = LL_i$



x_i

Security Measurement : Authenticated Key Exchange (AKE)

- No ideal-hash assumption
- Theorem

$$\mathrm{Adv}^{\mathrm{ake}}(t, Q, q_s) \leq 2nQ \cdot \mathrm{Adv}^{\mathrm{gddh}}(t') + 2 \cdot \mathrm{Adv}^{\mathrm{mddh}}(t) + n \cdot (n-1) \cdot \mathrm{Succ}^{\mathrm{cma}}(t) + \text{« negligible »}$$

$$t' \leq t + n \cdot Q \cdot T_{\mathrm{exp}}(k)$$

- Concepts of the proof
  - we define a sequence of games in an incremental way
  - we upper-bound the distance between the distributions of probability of two consecutive games
  - we finally combine these distances to upper-bound the probability of breaking the AKE security of the protocol

Defining the Games

- Game 0 : the adversary plays against the oracles in order to defeat the AKE security of the protocol
- Game 1 : we abort if the adversary produces a MAC forgery
- Game 2 : we simulate the protocol flows using the elements from a GDDH-tuple
- Game 3 : we simulate the protocol flows using the elements from a GDDH-tuple whose value $g^{x_1..x_n}$ is unknown
- Game 4 : we answer at random the Test-query and thus fix the adversary's probability of correctly guessing the bit b to be 1/2.

Outline

- Introduction
- Background
- Contributions
- Group DH key exchange
- Dynamic Group DH key exchange
- Refinements

> Practical aspects

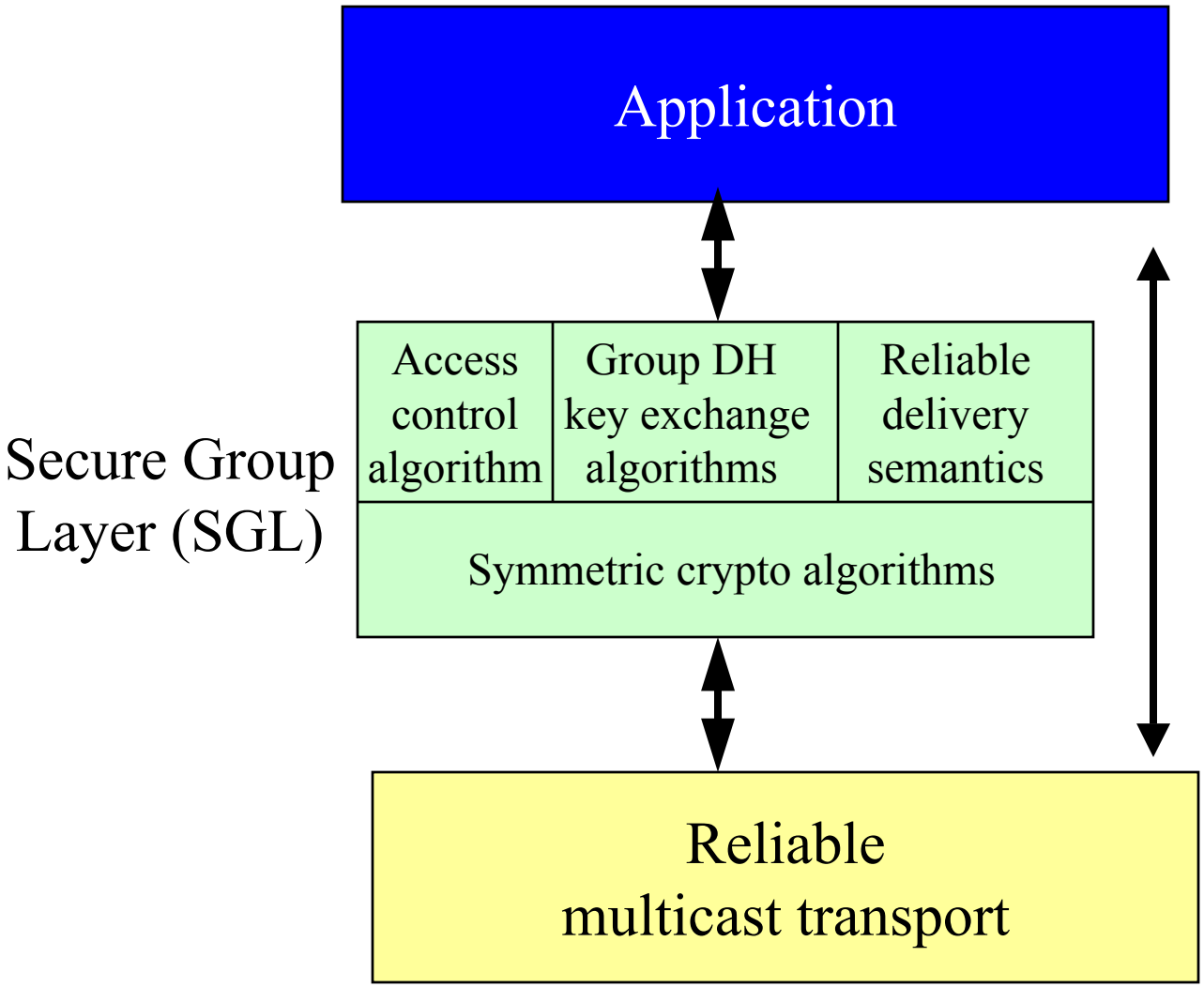
  — a security framework SGL to implement dynamic group DH key exchange

- Conclusion and further work

[ACTT01] Security Framework (SGL)

- An authenticated dynamic GDH key exchange algorithm enables group members to establish a session key
- A certificate-based access control mechanism makes sure that only the legitimate parties have access to the session key
  - — off-line (does not participate in key exchange)
- Symmetric crypto algorithms (e.g. Rijndael and HMAC)
  - — implement an authenticated and encrypted channel

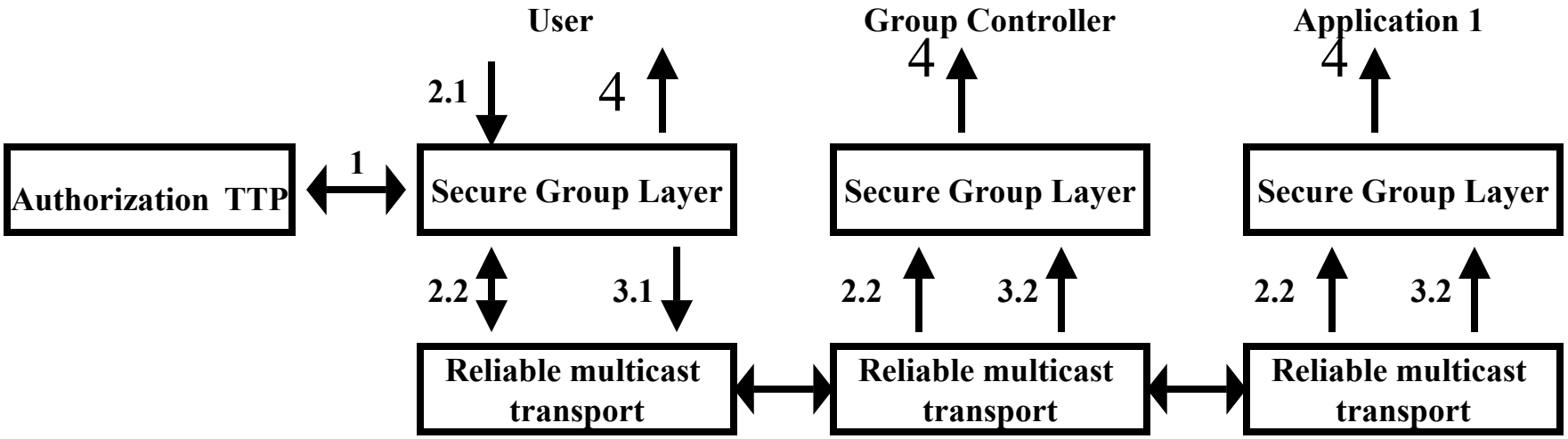
Secure and Reliable Multicast Communication Architecture



The Reliable Multicast Transport Layer

- Provide SGL with reliable and ordered delivery of messages
  - — data messages are delivered in order - FIFO, partial, and total - at each member of the group
- Provide SGL with membership notifications
  - — membership changes delivered in order with respect to data messages
- Several systems provide a reliable multicast layer
  - — e.g., Isis, Ensemble, Totem and InterGroup

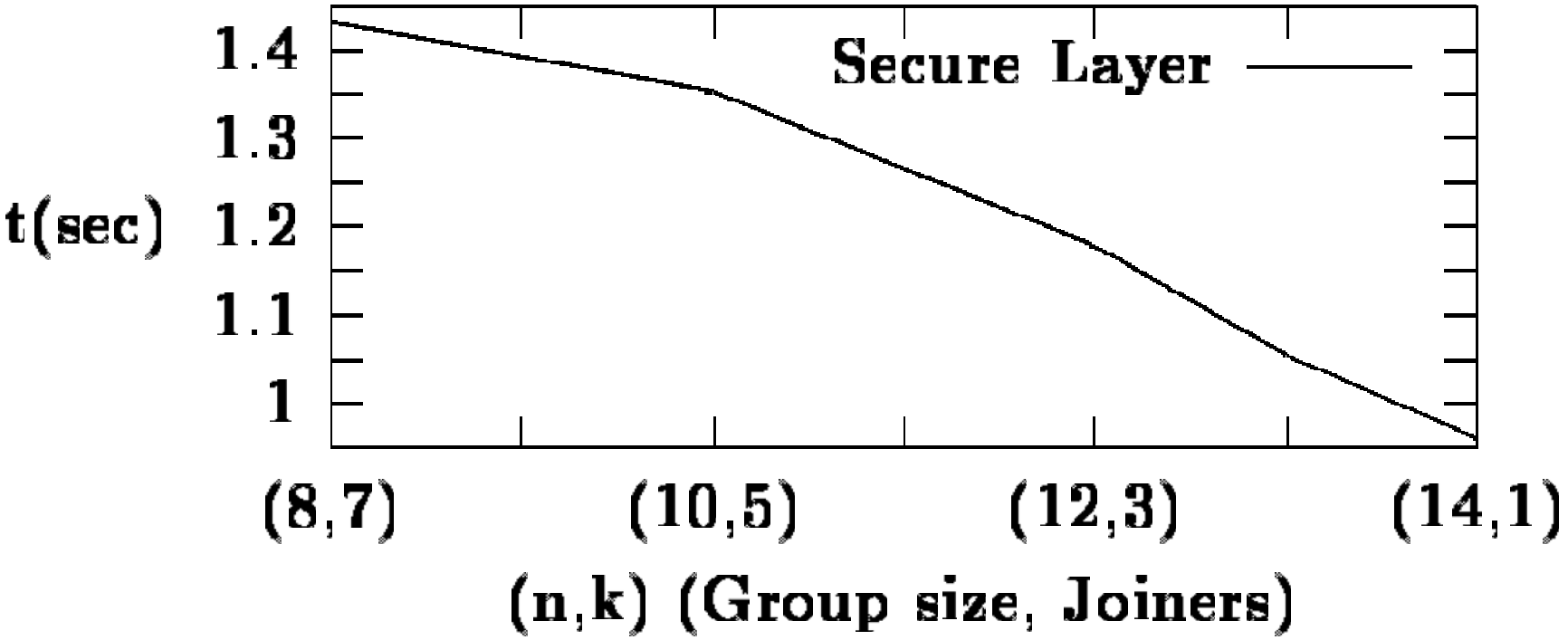
The Access Control Algorithm: a user join



1. **Authorization:** The user requests its permission from TTP and obtains a membership authorization certificate
2. **Join multicast group**:
   2.1.The user submits a join request
   2.2. Secure Group Layer gets a membership change notification
3. **Access control**:
   3.1. The user broadcasts its certificate
   3.2. GC checks the user's permission and, if authorized, initiates group DH key exchange
4. **Deliver secure membership**: When the group DH key exchange is done, Secure Group Layer delivers the secure membership notification to the application

A Preliminary Implementation of SGL

- Implementation in C : Totem, GDH with DSA, Akenti
- Performance : group size = 15 members, merge operation with variable-size sub-groups.



Conclusion

- Completed
  - [BCPQ01a] “Authenticated GDH key exchange: the static case”, ACM CCS’01
  - [BCP01b] “Authenticated GDH key exchange: the dynamic case”, Asiacrypt’01
  - [BCP02a] “Forward secrecy in GDH key exchange”, Eurocrypt’02
  - [ACTT01] “An Integrated Solution for Secure Group Communication in Wide-Area Networks”, IEEE Symposium on Computers and Communication’01
- Other related publications
  - [BCPPQ02] “Two Views of Authenticated GDH Key Exchange”, DIMACS Cryptographic Protocols in Complex Environments, 2002

Conclusion

- [BCP02b] “The Group Diffie-Hellman Problems”, SAC’02
- [BCP02c] “GDH Key Exchange secure against dictionary attacks”, submitted for publication to Asiacrypt’02
- [BAC02] “A Practical Approach to the InterGroup Protocols”, J. of Future Generation Computer Systems, 2002

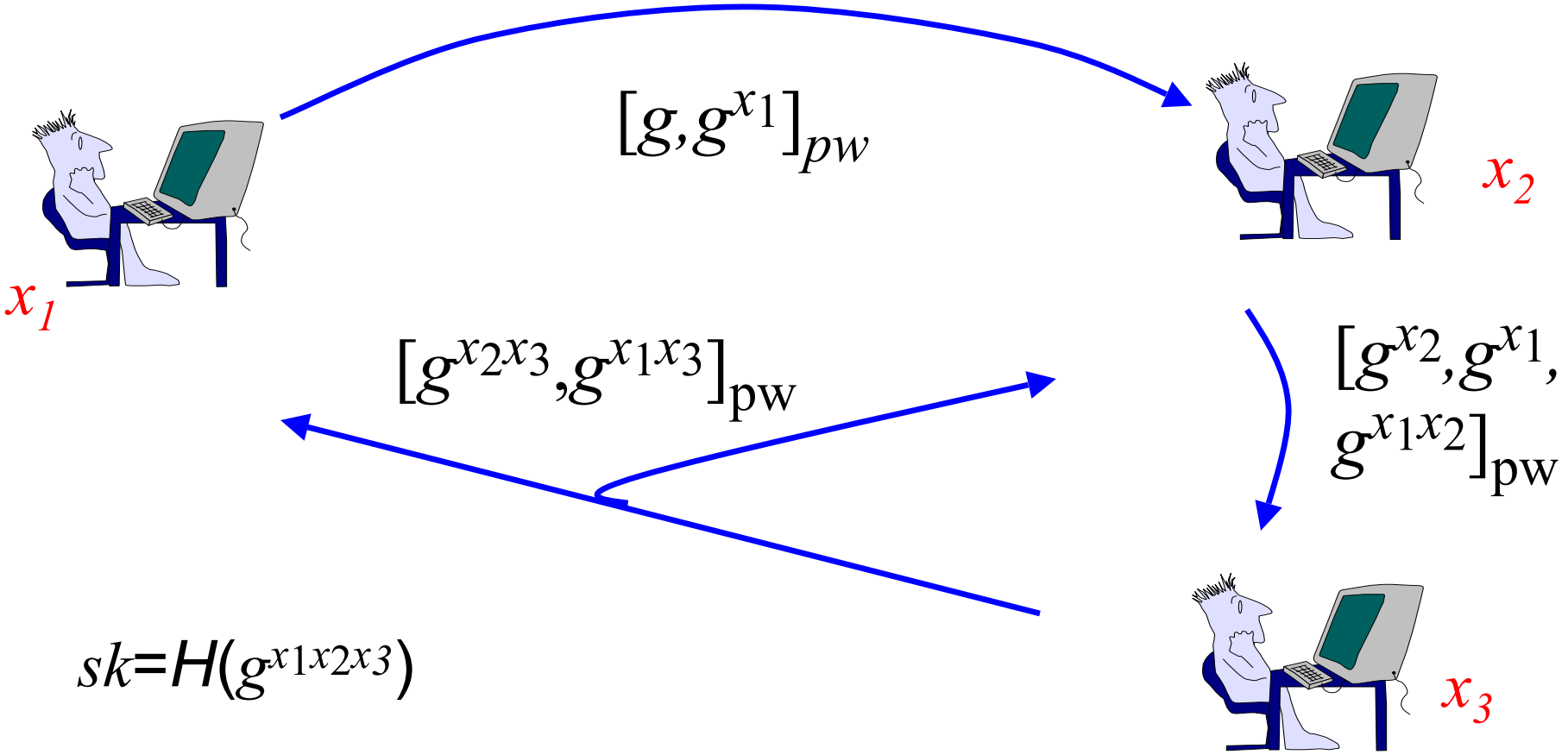
- Current and on-going work
  - SGL security improvements, interface definition and delivery semantics
  - Demonstration of an application using SGL and InterGroup

Appendix : Additional Security Goals

- Security against dictionary attacks
- Intractability assumptions
- Mutual Authentication (MA)

Password-Authenticated Group DH Exchange

- Security against dictionary attacks
- Algorithm: flows are encrypted using the password *pw*



Security Measurement : Dictionary Attacks

- Ideal-cipher assumption
- Theorem

$$\mathrm{Adv}^{ake}(T,q_s,q_e) \leq 2q_s/N + 2q_s \cdot (n\text{-}1) \cdot Adv^{ddh}(T') + 2 \cdot q_h \cdot \mathrm{Succ}^{tgcdh}(T') + \mathrm{Cte}$$

$$T' \leq T + n \cdot (3q_s + q_e) \cdot T_{exp}(k)$$

- The theorem shows that the security against dictionary attacks since the advantage of the adversary grows essentially with the ratio of interactions (number of send-queries) to the number of password.

- The security holds provided that DDH, TGCDH and M-DDH are hard. These terms can be made negligible.

Intractability Assumptions : Group Decisional Diffie-Hellman

- The DDH assumption
  - given the values g^{x_1}, g^{x_2}, one has to distinguish the value $g^{x_1 x_2}$ from a random one

- The DDH assumption generalized to the multi-party case
  - given *some* subsets of indices in $I=\{1,\ldots,n\}$ and all the values $g^{P_{i \in J} x_i}$ for every given subset J of I,
  - one has to distinguish the value $g^{x_1 \cdots x_n}$ from a random one

- Example with three parties ($n=3$ and $I=\{1,2,3\}$)
  - given the values g^{x_1}, g^{x_2}, g^{x_3} one has to distinguish the value $g^{x_1 x_2 x_3}$ from a random one

Intractability Assumptions : Multi-Decisional Diffie-Hellman

- The Multi-Decisional version of the DDH assumption (implied by DDH)
  - — given a set of values g^{x_i}, for i = 0, …,n
  - — one has to distinguish each of the values $g^{x_i x_j}$, 0< i < j< n, from a random one

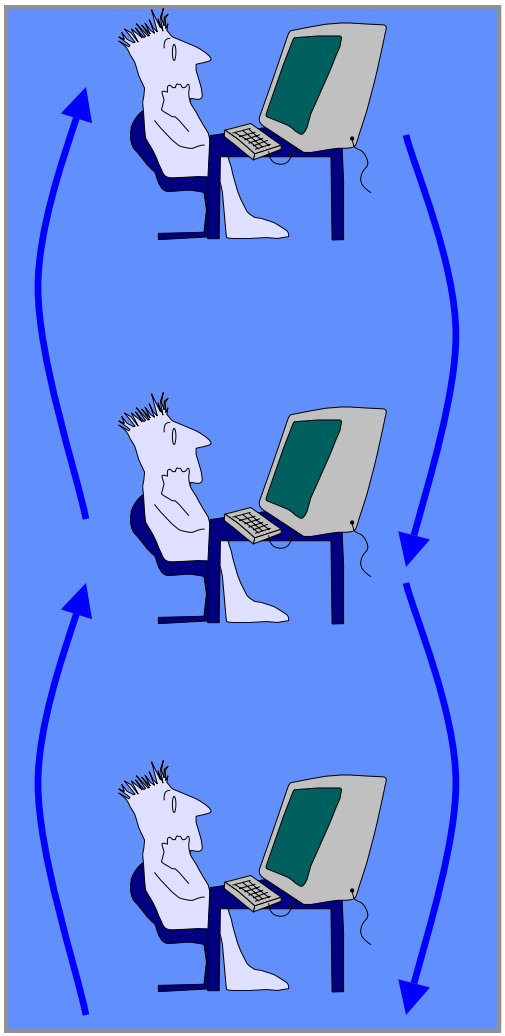
- Example with three parties *n=3*
  - — given the values g^{x_1}, g^{x_2}, g^{x_3}
  - — one has to distinguish each of the values $g^{x_1 x_2}$, $g^{x_1 x_3}$, $g^{x_2 x_3}$ from a random one

Security Goal : MA
Mutual Authentication

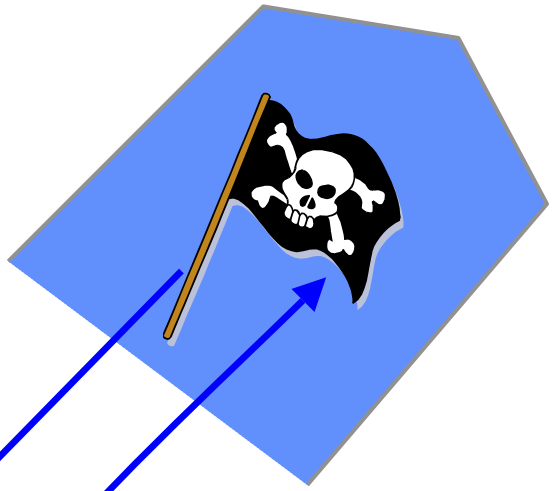
- Explicit authentication
  - — Each player is assured that his partners have actually computed the session key

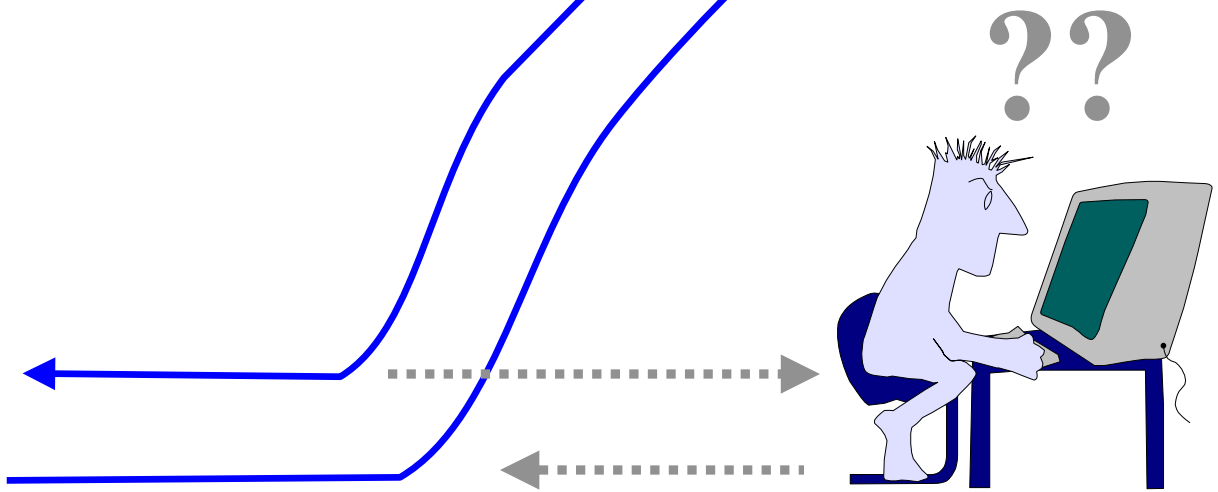
- Avoid impersonation attacks
  - — Only the legitimate partners are able to authenticate

Security Definitions (MA)



Adversary accepted as a partner by other players





A Protocol for Mutual Authenticated Key Exchange

- The session key is *sk*
- The algorithm :
  - U_i computes the authenticator $auth_i$ and brodcasts it

$$auth_i = H(\text{sk} \,||\, \text{i})$$

  - U_i computes the new session key sk as

$$sk' = H(sk \,||\, 0)$$

Security Measurement

- Ideal-hash assumption
- Theorem

$$Adv^{ake'}(T', q_s, q_h) \leq Adv^{ake}(T, q_s, q_h) + q_h / 2^l$$

$$Succ^{ma}(T', q_s, q_h) \leq Adv^{ake}(T, q_s, q_h) + n \cdot q_h / 2^l$$

$$T', T'' \leq T + (q_s + n \cdot T_{exp}(k))$$

- The adversary can break the protocols in two ways

  (1) the adversary is able to break the AKE security of protocol P' => it is possible to come up with an algorithm that break the AKE security of protocol P

  (2) the adversary is to able to break MA => it is possible to come up with an algo that break the AKE security of protocol P

Entropy-Smoothing Theorem

- Used to derivate keys from $g^{x_1..x_n}$
- Let D be a distribution of length s and entropy σ. Let H be a universal hash function from k-bits x s-bits to l-bits.

- Then the following $(l+k)$-bits distributions are $2^{-(e+1)}$-statistically close, where $l = \sigma - 2e$:

$$H_r(x)||r \quad \text{and} \quad y||r$$

$x \in_D \{0,1\}^s$, Uniform